```
                        PAGE    118,121
                        TITLE   WD1002S-WX2 Winchester Disk Controller BIOS (62-000042-012)

                        ;--------------------------------------------------------
                        ;       INT 13H Errors (returned in AH)                 :
                        ;--------------------------------------------------------

= 0001                  INT_13H_ERR_INVALID_COMMAND             EQU     001h
= 0002                  INT_13H_ERR_CANNOT_FIND_ADDR_MARK       EQU     002h
= 0004                  INT_13H_ERR_SECTOR_NOT_FOUND            EQU     004h
= 0005                  INT_13H_ERR_RESET_FAILED                EQU     005h
= 0007                  INT_13H_ERR_DRV_PARM_ACTIVITY_FAILED    EQU     007h
= 0009                  INT_13H_ERR_ATTEMPT_DMA_OVER_64K        EQU     009h
= 000B                  INT_13H_ERR_BAD_CYLINDER_OR_TRACK       EQU     00Bh
= 0010                  INT_13H_ERR_CRC_ECC_DATA                EQU     010h
= 0011                  INT_13H_ERR_ECC_CORRECTED_DATA          EQU     011h
= 0020                  INT_13H_ERR_CONTROLLER_FAILURE          EQU     020h
= 0040                  INT_13H_ERR_SEEK_FAILURE                EQU     040h
= 0080                  INT_13H_ERR_DRIVE_TIMED_OUT             EQU     080h
= 00BB                  INT_13H_ERR_UNDEFINED                   EQU     0BBh
= 00FF                  INT_13H_ERR_SENSE_FAILED                EQU     0FFh

                        ;--------------------------------------------------------
                        ;       Controller opcodes                              :
                        ;--------------------------------------------------------

= 0000                  OP_TEST_DRIVE_READY                     EQU     000h
= 0001                  OP_RECALIBRATE                          EQU     001h
= 0003                  OP_READ_SENSE                           EQU     003h
= 0004                  OP_FORMAT_DRIVE                         EQU     004h
= 0005                  OP_VERIFY_SECTORS                       EQU     005h
= 0006                  OP_FORMAT_TRACK                         EQU     006h
= 0007                  OP_FORMAT_BAD_TRACK                     EQU     007h
= 0008                  OP_READ_SECTORS                         EQU     008h
= 000A                  OP_WRITE_SECTORS                        EQU     00Ah
= 000B                  OP_SEEK                                 EQU     00Bh
= 000C                  OP_INIT_DRV_PARM                        EQU     00Ch
= 000D                  OP_READ_ECC_BURST_ERROR_LEN             EQU     00Dh
= 000E                  OP_READ_SECTOR_BUFFER                   EQU     00Eh
= 000F                  OP_WRITE_SECTOR_BUFFER                  EQU     00Fh
= 00E0                  OP_SECTOR_BUFFER_DIAG                   EQU     0E0h
= 00E3                  OP_DRIVE_DIAG                           EQU     0E3h
= 00E4                  OP_CTRL_DIAG                            EQU     0E4h
= 00E5                  OP_READ_LONG                            EQU     0E5h
= 00E6                  OP_WRITE_LONG                           EQU     0E6h

                        ;--------------------------------------------------------
                        ;       DMA mode register                               :
                        ;--------------------------------------------------------

= 0047                  DRQ3_READ       EQU     47H             ; channel=3 | address increment
= 004B                  DRQ3_WRITE      EQU     4BH             ;   | demand mode | read/write

                        ;--------------------------------------------------------
                        ;       Interrupt vector table (at 0000:0000)           :
                        ;--------------------------------------------------------

0000                    ZERO            SEGMENT AT 0H

= 0005                  IRQ5            EQU     5
0034                                    ORG     (8+IRQ5)*4
0034 ????               INT_IRQ5_OFF    DW      ?
0036 ????               INT_IRQ5_SEG    DW      ?

= 0013                  INT_13H         EQU     013H
004C                                    ORG     INT_13H*4
004C ????               INT_13H_OFF     DW      ?
004E ????               INT_13H_SEG     DW      ?

= 0019                  INT_19H         EQU     019H
0064                                    ORG     INT_19H*4
0064 ????               INT_19H_OFF     DW      ?
0066 ????               INT_19H_SEG     DW      ?

= 001E                  INT_1EH         EQU     01EH
0078                                    ORG     INT_1EH*4
0078 ????               INT_1EH_OFF     DW      ?
007A ????               INT_1EH_SEG     DW      ?

= 0040                  INT_40H         EQU     040H
0100                                    ORG     INT_40H*4
0100 ????               INT_40H_OFF     DW      ?
0102 ????               INT_40H_SEG     DW      ?

= 0041                  INT_41H         EQU     041H
0104                                    ORG     INT_41H*4
0104 ????               INT_41H_OFF     DW      ?
0106 ????               INT_41H_SEG     DW      ?

7C00                                    ORG     7C00H
7C00                    BOOT_SEC        LABEL   FAR
7DFE                                    ORG     7DFEh
7DFE                    BOOT_SIG        LABEL   FAR

7DFE                    ZERO            ends

                        ;--------------------------------------------------------
                        ;       BIOS Data Area (at 0040:0000)                   :
                        ;--------------------------------------------------------

0000                    BDA     SEGMENT AT 40H

0042                                    ORG     0042H           ; Buffer that holds Command Block
0042 ??                 SCRATCH         DB      ?               ;  and error sense
006C                                    ORG     006CH           ; Timer bumps this 18.2 times a
006C ????               TIMER_COUNTER   DW      ?               ;   second or so
0072                                    ORG     0072H
0072 ??                 SOFT_RESET_FLAG DB      ?
0074                                    ORG     0074H
0074 ??                 LAST_OP_STATUS  DB      ?
0075 ??                 NUM_OF_DRIVES   DB      ?

0076                    BDA             ends
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ;       Main ROM image                                  :
                        ;--------------------------------------------------------

0000                    ROM     segment byte public 'CODE'
                                assume  cs:ROM
                                assume  es:ZERO, ss:ZERO, ds:ZERO

0000 AA55                       dw      0AA55h                          ; Magic
0002 10                         db      16                              ; Length in 256-byte chunks
0003 EB 7C                      jmp     short near ptr ROM_ENTRY        ; Main option ROM entry point
0005 E9 0530 R                  jmp     DO_FORMAT                       ; Entry point to formatter

0008 28 43 29 20 43 6F          db      "(C) Copyright 1984 Western Digital Corporation"
     70 79 72 69 67 68
     74 20 31 39 38 34
     20 57 65 73 74 65
     72 6E 20 44 69 67
     69 74 61 6C 20 43
     6F 72 70 6F 72 61
     74 69 6F 6E

                                ; Yeah, do not disassemble or anything.
                                ; You'll be shot in your face if you do.

0036                    INT_1EH_TABLE:
0036 CF                         DB      0CFh            ; Bits 0-3: SRT step rate time
                                                        ; Bits 4-7: Head unload time
0037 02                         DB      2               ; Bit    0: 1=use DMA
                                                        ; Bits 2-7: Head load time
0038 25                         DB      37              ; 55-ms increments before turning disk motor off
0039 02                         DB      2               ; Sector size (0=128, 1=256, 2=512, 3=1024)
003A 08                         DB      8               ; EOT (last sector on a track)
003B 2A                         DB      42              ; Gap length for read/write operations
003C FF                         DB      255             ; Data Transfer Length
003D 50                         DB      80              ; Gap length for format operation
003E F6                         DB      0F6h            ; Fill character for format
003F 19                         DB      25              ; Head-settle time (in ms)
0040 04                         DB      4               ; Motor-startup time (in 1/8th-second intervals)

0041                    INT_41H_TABLE:
                                ; Microscience HH725
0041 0264                       DW      612             ; Cylinders
0043 04                         DB      4               ; Heads
0044 0265                       DW      613             ; RWC
0046 0265                       DW      613             ; WPC
0048 0B                         DB      0Bh             ; ECC Burst Length
0049 05                         DB      5               ; Control Byte
004A    07 [                    DB      7 DUP(0)
                00
                    ]


                                ; Miniscribe 3012
0051 0264                       DW      612             ; Cylinders
0053 02                         DB      2               ; Heads
0054 0080                       DW      128             ; RWC
0056 0080                       DW      128             ; WPC
0058 0B                         DB      0Bh             ; ECC Burst Length
0059 05                         DB      5               ; Control Byte
005A    07 [                    DB      7 DUP(0)
                00
                    ]


                                ; Seagate ST225
0061 0264                       DW      612             ; Cylinders
0063 04                         DB      4               ; Heads
0064 0265                       DW      613             ; RWC
0066 0080                       DW      128             ; WPC
0068 0B                         DB      0Bh             ; ECC Burst Length
0069 05                         DB      5               ; Control Byte
006A    07 [                    DB      7 DUP(0)
                00
                    ]


                                ; Seagate ST412
0071 0132                       DW      306             ; Cylinders
0073 04                         DB      4               ; Heads
0074 0132                       DW      306             ; RWC
0076 0000                       DW      0               ; WPC
0078 0B                         DB      0Bh             ; ECC Burst Length
0079 05                         DB      5               ; Control Byte
007A    07 [                    DB      7 DUP(0)
                00
                    ]


```

```
                        PAGE
                        ;--------------------------------------------------------
                        ;       Option ROM initialization                      :
                        ;--------------------------------------------------------

0081                    ROM_ENTRY       PROC    FAR
0081 33 C0                      xor     ax, ax
0083 8E D8                      mov     ds, ax

0085 FA                         cli
0086 C4 06 004C R               les     ax, dword ptr INT_13H_OFF
008A A3 0100 R                  mov     INT_40H_OFF, ax                 ; Saved floppy handler
008D 8C 06 0102 R               mov     INT_40H_SEG, es
0091 C7 06 004C R 01CF R        mov     INT_13H_OFF, offset INT_13H_HANDLER     ; Install disk handler
0097 8C 0E 004E R               mov     INT_13H_SEG, cs
009B C7 06 0064 R 014E R        mov     INT_19H_OFF, offset INT_19H_HANDLER     ; Install boot handler
00A1 8C 0E 0066 R               mov     INT_19H_SEG, cs
00A5 C7 06 0034 R 01BE R        mov     INT_IRQ5_OFF, offset IRQ_5_HANDLER      ; Install disk IRQ handler
00AB 8C 0E 0036 R               mov     INT_IRQ5_SEG, cs
00AF C7 06 0104 R 0041 R        mov     INT_41H_OFF, offset INT_41H_TABLE       ; Geometry table
00B5 8C 0E 0106 R               mov     INT_41H_SEG, cs
00B9 FB                         sti

00BA B8 ---- R                  mov     ax, BDA                         ; BIOS data area segment
00BD 8E D8                      mov     ds, ax
                                assume  ds:BDA

00BF C6 06 0075 R 00            mov     NUM_OF_DRIVES, 0
00C4 B8 019A                    mov     ax, 410                         ; Pre-initialize counter
00C7 81 3E 0072 R 1234          cmp     word ptr SOFT_RESET_FLAG, 1234h ;   to 22s for shorter (3s)
00CD 74 02                      je      short RESETTING                 ;   wait when resetting
00CF 33 C0                      xor     ax, ax                          ; Start from zero otherwise
00D1                    RESETTING:                                      ;   with plenty of time (25s)
00D1 A3 006C R                  mov     TIMER_COUNTER, ax               ;   for disk to spin up

00D4 FA                         cli
00D5 E4 21                      in      al, 21h
00D7 24 FE                      and     al, 0FEh                        ; Unmask IRQ0 (Timer)
00D9 E6 21                      out     21h, al
00DB FB                         sti

00DC B2 80                      mov     dl, 80h                         ; First hard drive
00DE                    DO_RESET:
00DE B9 0001                    mov     cx, 1
00E1 8A F5                      mov     dh, ch
00E3 8A C1                      mov     al, cl
00E5 B4 00                      mov     ah, 0                           ; Reset disk system
00E7 CD 13                      int     INT_13H
00E9 72 16                      jc      short ERROR_HAPPENED

00EB B4 14                      mov     ah, 14h                         ; Controller diagnostics
00ED CD 13                      int     INT_13H
00EF 72 10                      jc      short ERROR_HAPPENED

00F1                    WAIT_FOR_DRIVE_READY:
00F1 B4 10                      mov     ah, 10h                         ; Test drive ready
00F3 CD 13                      int     INT_13H
00F5 73 14                      jnc     short DRIVE_IS_READY

00F7 81 3E 006C R 01BE          cmp     TIMER_COUNTER, 446              ; Count up to 25 Seconds
00FD 72 F2                      jb      short WAIT_FOR_DRIVE_READY
00FF EB 27                      jmp     short TRY_AGAIN

0101                    ERROR_HAPPENED:
0101 FE C2                      inc     dl
0103 EB 23                      jmp     short TRY_AGAIN

0105                    STR_1701:
0105 31 37 30 31 0A 0D          db      "1701",0Ah,0Dh

010B                    DRIVE_IS_READY:
010B B4 11                      mov     ah, 11h                         ; Recalibrate drive
010D CD 13                      int     INT_13H
010F 72 17                      jc      short TRY_AGAIN

0111 FE 06 0075 R               inc     NUM_OF_DRIVES
0115 80 FA 80                   cmp     dl, 80h                         ; First hard drive?
0118 75 0E                      jne     short TRY_AGAIN

011A 81 3E 0072 R 1234          cmp     word ptr SOFT_RESET_FLAG, 1234h
0120 74 06                      je      short TRY_AGAIN
0122 C7 06 006C R 0165          mov     TIMER_COUNTER, 357              ; 20 seconds

0128                    TRY_AGAIN:
0128 FE C2                      inc     dl                              ; Next hard drive
012A F6 C2 01                   test    dl, 1
012D 75 C2                      jnz     short WAIT_FOR_DRIVE_READY
012F 80 FA 88                   cmp     dl, 88h                         ; Last hard drive?
0132 72 AA                      jb      short DO_RESET                  ; Not yet, reset another one
0134 80 3E 0075 R 00            cmp     NUM_OF_DRIVES, 0
0139 75 12                      jne     short RESET_SUCCESSFUL          ; At least one drive
                                                                        ;   initialized successfully
013B BE 0105 R                  mov     si, offset STR_1701             ; Otherwise proceed priniting
013E FC                         cld                                     ;   the 1701 error
013F B9 0006                    mov     cx, 6
0142                    PRINT_STRING:
0142 2E: AC                     lods    byte ptr cs:[si]
0144 B4 0E                      mov     ah, 0Eh                         ; TTY output
0146 CD 10                      int     10h
0148 E2 F8                      loop    PRINT_STRING
014A BD 000F                    mov     bp, 0Fh

014D                    RESET_SUCCESSFUL:
014D CB                         ret
014E                    ROM_ENTRY       ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ;       OS Boot Sector load service (INT 19H)           :
                        ;--------------------------------------------------------

014E                    INT_19H_HANDLER PROC    NEAR
014E 33 C0                      xor     ax, ax
0150 8E D8                      mov     ds, ax
                                assume  ds:ZERO
0152 8E C0                      mov     es, ax

0154 FA                         cli
0155 C7 06 0078 R 0036 R        mov     INT_1EH_OFF, offset INT_1EH_TABLE
015B 8C 0E 007A R               mov     INT_1EH_SEG, cs
015F C7 06 0104 R 0041 R        mov     INT_41H_OFF, offset INT_41H_TABLE
0165 8C 0E 0106 R               mov     INT_41H_SEG, cs
0169 FB                         sti

016A B9 0003                    mov     cx, 3                           ; Three attempts at floppy read
016D 33 D2                      xor     dx, dx                          ; Drive 0, head 0
016F                    READ_FLOPPY_BOOT_SEC:
016F B8 0000                    mov     ax, 0                           ; Reset Floppy disk system
0172 CD 40                      int     INT_40H
0174 72 0F                      jc      short FLOPPY_RESET_FAILED

0176 BB 7C00 R                  mov     bx, OFFSET BOOT_SEC
0179 B8 0201                    mov     ax, 201h                        ; AH=02H Read Sectors
                                                                        ; AL=01H One sector
017C 51                         push    cx
017D B9 0001                    mov     cx, 1
0180 CD 40                      int     INT_40H                         ; Read from floppy
0182 59                         pop     cx
0183 73 34                      jnc     short RUN_BOOT_SECTOR           ; Read successful!

0185                    FLOPPY_RESET_FAILED:
0185 80 FC 80                   cmp     ah, INT_13H_ERR_DRIVE_TIMED_OUT
0188 74 02                      je      short FLOPPY_BOOT_FAILED
018A E2 E3                      loop    READ_FLOPPY_BOOT_SEC

018C                    FLOPPY_BOOT_FAILED:
018C B8 0000                    mov     ax, 0                           ; Reset Floppy disk system
018F CD 40                      int     INT_40H

0191 B2 80                      mov     dl, 80h                         ; Start with first HDD
0193 B9 0008                    mov     cx, 8                           ; Up to eight hard disks
0196                    TRY_BOOT_NEXT_HD:
0196 B4 00                      mov     ah, 0                           ; Reset hard disk
0198 CD 13                      int     INT_13H
019A 72 17                      jc      short BOOT_FAILED

019C BB 7C00 R                  mov     bx, OFFSET BOOT_SEC
019F B8 0201                    mov     ax, 201h                        ; AH=02H Read Sectors
                                                                        ; AL=01H One sector
01A2 51                         push    cx
01A3 B9 0001                    mov     cx, 1
01A6 CD 13                      int     INT_13H
01A8 59                         pop     cx
01A9 72 08                      jc      short BOOT_FAILED

01AB 81 3E 7DFE R AA55          cmp     word ptr ds:BOOT_SIG, 0AA55h    ; See if magic matches
01B1 74 06                      je      short RUN_BOOT_SECTOR

01B3                    BOOT_FAILED:
01B3 FE C2                      inc     dl                              ; Try next disk
01B5 E2 DF                      loop    TRY_BOOT_NEXT_HD
01B7 CD 18                      int     18h                             ; ROM Basic, the epitome
                                                                        ;   of failure
01B9                    RUN_BOOT_SECTOR:
01B9 EA 7C00 ---- R             jmp     BOOT_SEC
01BE                    INT_19H_HANDLER ENDP

                        ;--------------------------------------------------------
                        ;       Physical disk interrupt (IRQ 5)                 :
                        ;--------------------------------------------------------

01BE                    IRQ_5_HANDLER   PROC    FAR
01BE 50                         push    ax

01BF B0 20                      mov     al, 20h                         ; ACK Interrupt
01C1 E6 20                      out     20h, al

01C3 B0 07                      mov     al, 7                           ; Mask DMA3
01C5 E6 0A                      out     0Ah, al

01C7 E4 21                      in      al, 21h
01C9 0C 20                      or      al, 20h                         ; Mask IRQ5
01CB E6 21                      out     21h, al

01CD 58                         pop     ax
01CE CF                         iret
01CF                    IRQ_5_HANDLER   ENDP

```

```
                        PAGE
                        ;--------------------------------------------------------
                        ;       BIOS Disk Services (INT 13H)                    :
                        ;--------------------------------------------------------

01CF                    INT_13H_HANDLER PROC    FAR
01CF FB                         sti
01D0 80 FA 80                   cmp     dl, 80h                         ; Is this a hard disk?
01D3 73 05                      jnb     short DO_HARD_DISK_INT          ;   Yes it is.
01D5 CD 40                      int     INT_40H                         ; Call original floppy handler,
01D7 CA 0002                    ret     2                               ;   if not.

01DA                    DO_HARD_DISK_INT:
01DA 80 FC 00                   cmp     ah, 0                           ; Is this a subsystem reset?
01DD 75 04                      jnz     short NOT_A_DISK_SUBSYS_RESET
01DF CD 40                      int     INT_40H                         ; Also reset floppies if it is
                                                                        ;   a disk subsystem reset.
01E1 B4 00                      mov     ah, 0                           ; Replace the return value
                                                                        ;   from floppy reset with
                                                                        ;   AH=0H reset.

01E3                    NOT_A_DISK_SUBSYS_RESET:
01E3 53                         push    bx
01E4 51                         push    cx
01E5 52                         push    dx
01E6 55                         push    bp
01E7 57                         push    di
01E8 56                         push    si
01E9 1E                         push    ds
01EA 06                         push    es
01EB 8B EC                      mov     bp, sp                          ; From here on, BP points
                                                                        ;   to the saved registers.

01ED 50                         push    ax
01EE B8 ---- R                  mov     ax, BDA
01F1 8E D8                      mov     ds, ax
                                assume  ds:BDA
01F3 58                         pop     ax

01F4 80 FC 14                   cmp     ah, offset INT_13H_NUM_OPS      ; We only understand 20 commands
01F7 77 05                      ja      short BAD_ARGUMENTS

01F9 80 FA 88                   cmp     dl, 88h                         ; We only support 8 hard disks
01FC 72 04                      jb      short GOOD_ARGUMENTS

01FE                    BAD_ARGUMENTS:
01FE B4 01                      mov     ah, INT_13H_ERR_INVALID_COMMAND
0200 EB 03                      jmp     short INT_13H_CLEANUP

0202                    GOOD_ARGUMENTS:
0202 E8 022B R                  call    INT_13H_DISPATCH_FUNCTION

0205                    INT_13H_CLEANUP:
0205 50                         push    ax
0206 88 26 0074 R               mov     LAST_OP_STATUS, ah

020A E8 0505 R                  call    PORT_323_TO_DX                  ; Write DMA/IRQ mask
020D B0 FC                      mov     al, 11111100b                   ; Disable IRQ and DMA
020F EE                         out     dx, al

0210 B0 07                      mov     al, 7                           ; Mask DRQ3
0212 E6 0A                      out     0Ah, al

0214 FA                         cli
0215 E4 21                      in      al, 21h
0217 0C 20                      or      al, 20h                         ; Mask IRQ5
0219 E6 21                      out     21h, al
021B FB                         sti

021C 80 C4 FF                   add     ah, 0FFh                        ; Why? Set overflow flag?
021F 58                         pop     ax
0220 07                         pop     es
0221 1F                         pop     ds
                                assume  ds:ZERO
0222 5E                         pop     si
0223 5F                         pop     di
0224 5D                         pop     bp
0225 5A                         pop     dx
0226 59                         pop     cx
0227 5B                         pop     bx
0228 CA 0002                    ret     2
022B                    INT_13H_HANDLER ENDP

```

```
                        PAGE
                        ;--------------------------------------------------------
                        ;       Handle hard disk services                       :
                        ;                                                       :
                        ; Fill in the Command Control Block and call the        :
                        ; appropriate handler for the function code.            :
                        ;                                                       :
                        ; Arguments:                                            :
                        ;   AL = Head number?                                   :
                        ;   DL = Drive                                          :
                        ;   CX = ???                                            :
                        ;   BP = Pointer to saved registers                     :
                        ;--------------------------------------------------------

022B                    INT_13H_DISPATCH_FUNCTION PROC  NEAR
                        assume  ds:BDA

022B A2 0046 R          mov     SCRATCH+4, al                   ; Sector Count

022E FE C9              dec     cl
0230 89 0E 0044 R       mov     word ptr SCRATCH+2, cx          ; Cylinder | Sector

0234 8A EA              mov     ch, dl
0236 80 E5 01           and     ch, 1
0239 B1 05              mov     cl, 5
023B D2 E5              shl     ch, cl
023D 80 E6 0F           and     dh, 0Fh
0240 0A EE              or      ch, dh
0242 88 2E 0043 R       mov     SCRATCH+1, ch                   ; Drive | Head

0246 80 EA 80           sub     dl, 80h
0249 81 E2 00FE         and     dx, 0FEh
024D D0 E2              shl     dl, 1
024F 8B F2              mov     si, dx
0251 50                 push    ax
0252 06                 push    es
0253 E8 04D9 R          call    GET_DRIVE_PARMS
0256 26: 8A 47 08       mov     al, es:[bx+8]                   ; Copy command byte from drive params
025A A2 0047 R          mov     SCRATCH+5, al                   ; R1 R2 0 0 0 SP SP SP (Command block)
025D 07                 pop     es
025E 58                 pop     ax

025F 8A C4              mov     al, ah
0261 BB 029E R          mov     bx, offset INT_13H_OPS
0264 2E: D7             xlat    byte ptr cs:[bx]
0266 A2 0042 R          mov     SCRATCH+0, al                   ; Command block: OP CODE

0269 8A DC              mov     bl, ah                          ; Now that the Command Block is
026B 32 FF              xor     bh, bh                          ;   constructed, pick a function
026D D0 E3              shl     bl, 1                           ;   to submit the command
026F 2E: FF A7 0274 R   jmp     cs:INT_13H_DISP[bx]
0274                    INT_13H_DISPATCH_FUNCTION       ENDP

0274 02B3 R             INT_13H_DISP    dw offset HANDLE_DISK_RESET     ; 00H Reset disk system
0276 035C R                             dw offset HANDLE_GET_LAST_STAT  ; 01H Get status of last operation
0278 037E R                             dw offset HANDLE_READ_SECTORS   ; 02H Read sectors
027A 0361 R                             dw offset HANDLE_WRITE_SECTORS  ; 03H Write sectors
027C 03D5 R                             dw offset HANDLE_NO_DMA         ; 04H Verify sectors
027E 03D5 R                             dw offset HANDLE_NO_DMA         ; 05H Format track
0280 03D5 R                             dw offset HANDLE_NO_DMA         ; 06H Format bad track
0282 03D5 R                             dw offset HANDLE_NO_DMA         ; 07H Format drive from given track
0284 0335 R                             dw offset HANDLE_GET_DRV_PARM   ; 08H Read disk parameters
0286 02D5 R                             dw offset HANDLE_INIT_DRV_PARM  ; 09H Initialize disk controller
0288 0365 R                             dw offset HANDLE_READ_LONG      ; 0AH Read long sectors
028A 0370 R                             dw offset HANDLE_WRITE_LONG     ; 0BH Write long sectors
028C 03D5 R                             dw offset HANDLE_NO_DMA         ; 0CH Seek
028E 02B3 R                             dw offset HANDLE_DISK_RESET     ; 0DH Reset disk drives
0290 0374 R                             dw offset HANDLE_READ_RAM       ; 0EH Read sector buffer
0292 037A R                             dw offset HANDLE_WRITE_RAM      ; 0FH Write sector buffer
0294 03D5 R                             dw offset HANDLE_NO_DMA         ; 10H Test whether drive is ready
0296 03D5 R                             dw offset HANDLE_NO_DMA         ; 11H Recalibrate
0298 03D5 R                             dw offset HANDLE_NO_DMA         ; 12H Sector buffer diagnostics
029A 03D5 R                             dw offset HANDLE_NO_DMA         ; 12H Drive diagnostics
029C 03D5 R                             dw offset HANDLE_NO_DMA         ; 13H Controller diagnostics

029E 0C                 INT_13H_OPS     db OP_INIT_DRV_PARM
029F 00                                 db OP_TEST_DRIVE_READY
02A0 08                                 db OP_READ_SECTORS
02A1 0A                                 db OP_WRITE_SECTORS
02A2 05                                 db OP_VERIFY_SECTORS
02A3 06                                 db OP_FORMAT_TRACK
02A4 07                                 db OP_FORMAT_BAD_TRACK
02A5 04                                 db OP_FORMAT_DRIVE
02A6 00                                 db OP_TEST_DRIVE_READY
02A7 0C                                 db OP_INIT_DRV_PARM
02A8 E5                                 db OP_READ_LONG
02A9 E6                                 db OP_WRITE_LONG
02AA 0B                                 db OP_SEEK
02AB 0C                                 db OP_INIT_DRV_PARM
02AC 0E                                 db OP_READ_SECTOR_BUFFER
02AD 0F                                 db OP_WRITE_SECTOR_BUFFER
02AE 00                                 db OP_TEST_DRIVE_READY
02AF 01                                 db OP_RECALIBRATE
02B0 E0                                 db OP_SECTOR_BUFFER_DIAG
02B1 E3                                 db OP_DRIVE_DIAG
= 0014                  INT_13H_NUM_OPS EQU     $-INT_13H_OPS
02B2 E4                                 db OP_CTRL_DIAG
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ;       Disk reset (AH=00H)                             :
                        ;--------------------------------------------------------

02B3                    HANDLE_DISK_RESET PROC  NEAR
02B3 E8 0500 R                  call    PORT_321_TO_DX
02B6 EE                         out     dx, al                          ; Assert RESET

02B7 B9 0584                    mov     cx, 1412                        ; Wait while, hopefully RESET
02BA E2 FE              WAIT_A_BIT:     loop    WAIT_A_BIT              ;   will get asserted

02BC B4 0A                      mov     ah, 10                          ; Ten attempts
02BE                    WAIT_BUSY:
02BE 42                         inc     dx                              ; Port 0x322 now
02BF EE                         out     dx, al                          ; Assert BUSY

02C0 4A                         dec     dx                              ; Back to port 0x321
02C1 EC                         in      al, dx                          ; Read STATUS

02C2 A8 30                      test    al, 110000b                     ; IRQ | DRQ
02C4 75 0C                      jne     short BAD_STATUS

02C6 24 0D                      and     al, 1101b                       ; BUSY | C/D | REQ
02C8 34 0D                      xor     al, 1101b
02CA 74 09                      je      short HANDLE_INIT_DRV_PARM      ; Reset succeeded. Proceed
                                                                        ;       configuring the drives.
02CC E2 F0                      loop    WAIT_BUSY                       ; What? Isn't CX=0 now?

02CE FE CC                      dec     ah                              ; More attempts?
02D0 75 EC                      jnz     short WAIT_BUSY                 ;   Yes. Wait some more.

                                ; FALL THROUGH TO BAD_STATUS
02D2                    HANDLE_DISK_RESET       ENDP

02D2                    BAD_STATUS:
02D2 B4 05                      mov     ah, INT_13H_ERR_RESET_FAILED
02D4                    JUST_RETURN_0:
02D4 C3                         ret

                        ;--------------------------------------------------------
                        ;       Initialize disk parameters (AH=09H)             :
                        ;                                                       :
                        ; Also called upon successful disk reset (AH=00H)       :
                        ;--------------------------------------------------------

02D5                    HANDLE_INIT_DRV_PARM PROC       NEAR
02D5 C6 06 0043 R 00            mov     SCRATCH+1, 0                    ; Select first drive
02DA E8 02E4 R                  call    INIT_DRV_PARM                   ; Initialize first drive
02DD 72 F5                      jc      short JUST_RETURN_0
02DF C6 06 0043 R 20            mov     SCRATCH+1, 20h                  ; Select second drive
02E4                    INIT_DRV_PARM:
02E4 B0 FC                      mov     al, 11111100b                   ; Disable IRQ, DRQ
02E6 E8 0479 R                  call    WRITE_COMMAND_BLOCK
02E9 72 E9                      jc      short JUST_RETURN_0
02EB E8 04D9 R                  call    GET_DRIVE_PARMS                 ; ES:BX now points to params table

02EE BF 0001                    mov     di, 1                           ; Write the seven bytes
02F1 E8 0328 R                  call    WRITE_DATA                      ;   of drive paramerets
02F4 BF 0000                    mov     di, 0                           ;   from ES:[BX+DI]
02F7 E8 0328 R                  call    WRITE_DATA
02FA BF 0002                    mov     di, 2
02FD E8 0328 R                  call    WRITE_DATA
0300 BF 0004                    mov     di, 4
0303 E8 0328 R                  call    WRITE_DATA
0306 BF 0003                    mov     di, 3
0309 E8 0328 R                  call    WRITE_DATA
030C BF 0006                    mov     di, 6
030F E8 0328 R                  call    WRITE_DATA
0312 BF 0005                    mov     di, 5
0315 E8 0328 R                  call    WRITE_DATA
0318 BF 0007                    mov     di, 7
031B E8 0328 R                  call    WRITE_DATA

031E E8 04AB R                  call    CHECK_STATUS
0321 72 02                      jc      short RETURN_ERR_7
0323 74 02                      jz      short JUST_RETURN_3             ; Success!

0325                    RETURN_ERR_7:
0325 B4 07                      mov     ah, INT_13H_ERR_DRV_PARM_ACTIVITY_FAILED
0327                    JUST_RETURN_3:
0327 C3                         ret                                     ; Returns into this procedure
                                                                        ;   (above) on first drive,
                                                                        ;   returns from procedure
                                                                        ;   on second drive.
0328                    WRITE_DATA:
0328 E8 04C8 R                  call    WAIT_FOR_REQ
032B 72 05                      jc      short RETURN_FROM_CALLER

032D 26: 8A 01                  mov     al, es:[bx+di]
0330 EE                         out     dx, al
0331 C3                         ret

0332                    RETURN_FROM_CALLER:
0332 58                         pop     ax                              ; Throw away the intra-procedure
                                                                        ;   return value, returning directly
                                                                        ;   to this procedure's caller.
0333 EB F0                      jmp     short RETURN_ERR_7
0335                    HANDLE_INIT_DRV_PARM    ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ;       Get disk parameters (AH=08H)                    :
                        ;--------------------------------------------------------

0335                    HANDLE_GET_DRV_PARM PROC        NEAR
0335 E8 04D9 R                  call    GET_DRIVE_PARMS

0338 26: 8B 07                  mov     ax, es:[bx]
033B 2D 0002                    sub     ax, 2
033E 8A E8                      mov     ch, al
0340 D1 E8                      shr     ax, 1
0342 D1 E8                      shr     ax, 1
0344 24 C0                      and     al, 0C0h
0346 0C 11                      or      al, 11h
0348 8A E5                      mov     ah, ch                          ; Cylinders & tracks to
034A 89 46 0C                   mov     [bp+0Ch], ax                    ;   saved CX

034D 26: 8A 67 02               mov     ah, es:[bx+2]
0351 FE CC                      dec     ah
0353 A0 0075 R                  mov     al, NUM_OF_DRIVES               ; Heads & drives to
0356 89 46 0A                   mov     [bp+0Ah], ax                    ;   saved DX

                                ; FALL THROUGH TO RETURN_ZERO
0359                    HANDLE_GET_DRV_PARM     ENDP

0359                    RETURN_ZERO:
0359 B4 00                      mov     ah, 0
035B C3                         ret

                        ;--------------------------------------------------------
                        ;       Get status of last operation (AH=01H)           :
                        ;--------------------------------------------------------

035C                    HANDLE_GET_LAST_STAT PROC       NEAR
035C A0 0074 R                  mov     al, LAST_OP_STATUS
035F EB F8                      jmp     short RETURN_ZERO
0361                    HANDLE_GET_LAST_STAT    ENDP

                        ;--------------------------------------------------------
                        ;       Write sectors (AH=03H)                          :
                        ;--------------------------------------------------------

0361                    HANDLE_WRITE_SECTORS PROC       NEAR
0361 B0 4B                      mov     al, DRQ3_WRITE
0363 EB 1B                      jmp     short COMMON_DMA_SECTORS_512
0365                    HANDLE_WRITE_SECTORS    ENDP

                        ;--------------------------------------------------------
                        ;       Read long (with ECC) sectors (AH=0AH)           :
                        ;--------------------------------------------------------

0365                    HANDLE_READ_LONG PROC   NEAR
0365 B0 47                      mov     al, DRQ3_READ
                                ; FALL THROUGH TO COMMON_DMA_SECTORS_516
0367                    HANDLE_READ_LONG        ENDP

0367                    COMMON_DMA_SECTORS_516 PROC     NEAR
0367 8A 16 0046 R               mov     dl, SCRATCH+4
036B BF 0204                    mov     di, 516                 ; 516 bytes = 512 bytes + 4 ECC
036E EB 17                      jmp     short COMMON_DMA
0370                    COMMON_DMA_SECTORS_516  ENDP

                        ;--------------------------------------------------------
                        ;       Write long (with ECC) sectors (AH=0BH)          :
                        ;--------------------------------------------------------

0370                    HANDLE_WRITE_LONG PROC  NEAR
0370 B0 4B                      mov     al, DRQ3_WRITE
0372 EB F3                      jmp     short COMMON_DMA_SECTORS_516
0374                    HANDLE_WRITE_LONG       ENDP

                        ;--------------------------------------------------------
                        ;       Read sector buffer (AH=0EH)                     :
                        ;--------------------------------------------------------

0374                    HANDLE_READ_RAM PROC    NEAR
0374 B0 47                      mov     al, DRQ3_READ
                                ; FALL THROUGH TO COMMON_READ_WRITE_RAM
0376                    HANDLE_READ_RAM ENDP

0376                    COMMON_READ_WRITE_RAM PROC      NEAR
0376 B2 01                      mov     dl, 1
0378 EB 0A                      jmp     short COMMON_DMA_512
037A                    COMMON_READ_WRITE_RAM   ENDP

                        ;--------------------------------------------------------
                        ;       Write sector buffer (AH=0FH)                    :
                        ;--------------------------------------------------------

037A                    HANDLE_WRITE_RAM PROC   NEAR
037A B0 4B                      mov     al, DRQ3_WRITE
037C EB F8                      jmp     short COMMON_READ_WRITE_RAM
037E                    HANDLE_WRITE_RAM        ENDP

                        ;--------------------------------------------------------
                        ;       Read sectors (AH=02H)                           :
                        ;--------------------------------------------------------

037E                    HANDLE_READ_SECTORS PROC        NEAR
037E B0 47                      mov     al, DRQ3_READ
                                ; FALL THROUGH TO COMMON_DMA_SECTORS_512
0380                    HANDLE_READ_SECTORS     ENDP

0380                    COMMON_DMA_SECTORS_512      PROC        NEAR
0380 8A 16 0046 R               mov     dl, SCRATCH+4
                                ; FALL THROUGH TO COMMON_DMA_512
0384                    COMMON_DMA_SECTORS_512  ENDP

0384                    COMMON_DMA_512          PROC    NEAR
0384 BF 0200                    mov     di, 512
                                ; FALL THROUGH TO COMMON_DMA
0387                    COMMON_DMA_512          ENDP

```

```
                        PAGE
                        ;--------------------------------------------------------
                        ; Arguments:                                            :
                        ;   AL    = DMA mode register (DRQ3_WRITE/DRQ3_READ)    :
                        ;   ES:BX = Buffer                                       :
                        ;   BP    = Saved regs                                  :
                        ;   DL    = Block Count (on other entry point)          :
                        ;   DI    = 512/516 (transfer length)                   :
                        ;--------------------------------------------------------

0387                    COMMON_DMA              PROC    NEAR
0387 FA                         cli
0388 E6 0B                      out     0Bh, al                         ; Set DMA mode register
038A E6 0C                      out     0Ch, al                         ; Clear DMA byte pointer

038C 8C C0                      mov     ax, es
038E B1 04                      mov     cl, 4
0390 D3 C0                      rol     ax, cl
0392 8A E8                      mov     ch, al
0394 24 F0                      and     al, 0F0h
0396 03 46 0E                   add     ax, [bp+0Eh]                    ; Saved AX???
0399 80 D5 00                   adc     ch, 0
039C E6 06                      out     6, al                           ; DRQ3 base address
                                                                        ;   (also sets current address)
039E 86 E0                      xchg    ah, al
03A0 E6 06                      out     6, al                           ; DRQ3 base address
                                                                        ;  (also sets current address)
03A2 86 C5                      xchg    al, ch
03A4 8A CC                      mov     cl, ah
03A6 24 0F                      and     al, 0Fh
03A8 E6 82                      out     82h, al                         ; DRQ1 address bits 16-23 latch
03AA 8B C7                      mov     ax, di
03AC 32 F6                      xor     dh, dh
03AE F7 E2                      mul     dx
03B0 2D 0001                    sub     ax, 1
03B3 80 DA 00                   sbb     dl, 0
03B6 E6 07                      out     7, al                           ; DRQ3 base address and word count
03B8 86 C4                      xchg    al, ah
03BA E6 07                      out     7, al                           ; DRQ3 base address and word count
03BC FB                         sti

03BD 75 13                      jnz     short BAD_DMA                   ; Check DMA address validity
03BF 86 E0                      xchg    ah, al
03C1 03 C1                      add     ax, cx
03C3 72 0D                      jb      short BAD_DMA

03C5 B0 03                      mov     al, 11b                         ; Enable IRQ and DMA
03C7 E8 0479 R                  call    WRITE_COMMAND_BLOCK
03CA 72 08                      jc      short JUST_RETURN_2

03CC B0 03                      mov     al, 11b                         ; Mask DRQ 3
03CE E6 0A                      out     0Ah, al

03D0 EB 0A                      jmp     short HANDLE_COMMON             ; Jump

03D2                    BAD_DMA:
03D2 B4 09                      mov     ah, INT_13H_ERR_ATTEMPT_DMA_OVER_64K
                                ; FALL THROUGH TO JUST_RETURN_2
03D4                    COMMON_DMA      ENDP

03D4                    JUST_RETURN_2:
03D4 C3                         ret

03D5                    HANDLE_NO_DMA   PROC    NEAR
03D5 B0 02                      mov     al, 2
03D7 E8 0479 R                  call    WRITE_COMMAND_BLOCK
03DA 72 F8                      jb      short JUST_RETURN_2
                                ; FALL THROUGH TO HANDLE_COMMON
03DC                    HANDLE_NO_DMA   ENDP


```

```
                        PAGE
03DC                    HANDLE_COMMON   PROC    NEAR
03DC FA                         cli
03DD E4 21                      in      al, 21h
03DF 24 DF                      and     al, 0DFh                ; Unmask IRQ5
03E1 E6 21                      out     21h, al
03E3 FB                         sti

03E4 E8 0500 R                  call    PORT_321_TO_DX
03E7                    WAIT_FOR_IRQ:
03E7 EC                         in      al, dx                  ; Read Status
03E8 A8 20                      test    al, 100000b             ; IRQ
03EA 75 07                      jnz     short GOT_IRQ
03EC A8 08                      test    al, 1000b               ; BUSY
03EE 75 F7                      jnz     short WAIT_FOR_IRQ

03F0                    RETURN_BAD_STATUS:
03F0 E9 048A R                  jmp     RETURN_80

03F3                    GOT_IRQ:
03F3 E8 0505 R                  call    PORT_323_TO_DX          ; Write DMA/IRQ mask
03F6 B0 FC                      mov     al, 0FCh                ; Disable IRQ and DMA
03F8 EE                         out     dx, al

03F9 E8 04AB R                  call    CHECK_STATUS
03FC 72 F2                      jc      short RETURN_BAD_STATUS
03FE 74 D4                      jz      short JUST_RETURN_2

0400 C6 06 0042 R 03            mov     SCRATCH+0, OP_READ_SENSE
0405 B0 FC                      mov     al, 0FCh                ; Disable IRQ and DMA
0407 E8 0479 R                  call    WRITE_COMMAND_BLOCK
040A 72 6A                      jc      short RETURN_FF_ERROR   ; Jump if Below (CF=1)

040C BF 0042 R                  mov     di, offset SCRATCH      ; Read sense bytes (4)
040F 8C D8                      mov     ax, ds                  ;   into scratch block
0411 8E C0                      mov     es, ax
                                assume  es:BDA
0413 B9 0004                    mov     cx, 4                   ; Four bytes of sense
0416 FC                         cld
0417                    READ_SENSE_BYTE:
0417 E8 04C8 R                  call    WAIT_FOR_REQ
041A 72 5A                      jc      short RETURN_FF_ERROR
041C EC                         in      al, dx
041D AA                         stosb
041E E2 F7                      loop    READ_SENSE_BYTE

0420 E8 04AB R                  call    CHECK_STATUS            ; See if we could read sense
0423 72 51                      jc      short RETURN_FF_ERROR   ; Not at all!
0425 75 4F                      jnz     short RETURN_FF_ERROR

0427 8A 2E 0042 R               mov     ch, SCRATCH+0           ; Sense error code
042B 8A DD                      mov     bl, ch                  ; SENSE_TO_ERR is a sparse
042D 81 E3 0030                 and     bx, 30h                 ;   error mapping table.
0431 B1 03                      mov     cl, 3                   ;   First bits 4-5 are used
0433 D2 EB                      shr     bl, cl                  ;   as an index where range
0435 8A E5                      mov     ah, ch                  ;   and limits are found.
0437 80 E4 0F                   and     ah, 0Fh                 ;   Then bits 0-3 are used
043A 2E: 3A A7 050A R           cmp     ah, cs:SENSE_TO_ERR[bx] ; to index the actual
043F 73 32                      jnb     short RETURN_BB         ;   value (that is the
0441 43                         inc     bx                      ;   INT 13h error code).
0442 2E: 8A 9F 050A R           mov     bl, cs:SENSE_TO_ERR[bx] ; Bounds are checked along
0447 02 DC                      add     bl, ah                  ; the way
0449 2E: 8A A7 050A R           mov     ah, cs:SENSE_TO_ERR[bx]

044E 80 FD 18                   cmp     ch, 18h                 ; Error Code 18H is a special
                                                                ;   case, handled below:
                                                                ;   Correctable Data Error
0451 75 22                      jne     short JUST_RETURN_1     ; We're done unless it's 18H

0453 8A FC                      mov     bh, ah
0455 C6 06 0042 R 0D            mov     SCRATCH+0, OP_READ_ECC_BURST_ERROR_LEN
045A B0 FC                      mov     al, 0FCh                ; Disable IRQ and DMA
045C E8 0479 R                  call    WRITE_COMMAND_BLOCK
045F 72 15                      jc      short RETURN_FF_ERROR
0461 E8 04C8 R                  call    WAIT_FOR_REQ
0464 72 10                      jc      short RETURN_FF_ERROR

0466 EC                         in      al, dx                  ; Read Status???
0467 8A D8                      mov     bl, al

0469 E8 04AB R                  call    CHECK_STATUS
046C 72 08                      jc      short RETURN_FF_ERROR
046E 75 06                      jnz     short RETURN_FF_ERROR

0470 8B C3                      mov     ax, bx
0472 C3                         ret
0473                    RETURN_BB:
0473 B4 BB                      mov     ah, INT_13H_ERR_UNDEFINED
0475                    JUST_RETURN_1:
0475 C3                         ret
0476                    RETURN_FF_ERROR:
0476 B4 FF                      mov     ah, INT_13H_ERR_SENSE_FAILED
0478 C3                         ret
0479                    HANDLE_COMMON   ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ; Parameters:                                           :
                        ;   AL = IRQ/DRQ mask register (323H)                   :
                        ;--------------------------------------------------------

0479                    WRITE_COMMAND_BLOCK PROC        NEAR
0479 E8 0505 R                  call    PORT_323_TO_DX
047C EE                         out     dx, al                          ; Write DMA/IRQ mask

047D 4A                         dec     dx                              ; Port now 0x322
047E EE                         out     dx, al                          ; Enable BUSY and prepare for command

047F 4A                         dec     dx                              ; Port now 0x321
0480 B9 012C                    mov     cx, 300                         ; 300 Tries
0483                    WAIT_UNTIL_BUSY:
0483 EC                         in      al, dx                          ; Read Status
0484 A8 08                      test    al, 1000b                       ; BUSY
0486 75 06                      jnz     short BUSY
0488 E2 F9                      loop    WAIT_UNTIL_BUSY                 ; Read Status

048A                    RETURN_80:
048A B4 80                      mov     ah, INT_13H_ERR_DRIVE_TIMED_OUT
048C F9                         stc
048D                    JUST_RETURN_4:
048D C3                         ret

048E                    BUSY:
048E BF 0042 R                  mov     di, offset SCRATCH+0            ; Command block: OP CODE
0491 B9 0006                    mov     cx, 6                           ; Command Control Block has 6 bytes
0494 FC                         cld

0495                    WRITE_COMMAND_BYTE:
0495 E8 04C8 R                  call    WAIT_FOR_REQ
0498 72 F3                      jc      short JUST_RETURN_4
049A 24 0E                      and     al, 1110b
049C 34 0C                      xor     al, 1100b
049E 75 EA                      jne     short RETURN_80
04A0 87 FE                      xchg    di, si
04A2 AC                         lodsb
04A3 87 FE                      xchg    di, si
04A5 EE                         out     dx, al
04A6 E2 ED                      loop    WRITE_COMMAND_BYTE

04A8 E9 0359 R                  jmp     RETURN_ZERO
04AB                    WRITE_COMMAND_BLOCK     ENDP

                        ;--------------------------------------------------------
                        ; This waits for command to finish, reads the Command   :
                        ; Completion Byte and checks the error bit.             :
                        ;                                                       :
                        ; Result:                                               :
                        ;   CF = 0 No error                                     :
                        ;   CF = 1 Error                                        :
                        ;--------------------------------------------------------

04AB                    CHECK_STATUS    PROC    NEAR
04AB E8 04C8 R                  call    WAIT_FOR_REQ
04AE B4 00                      mov     ah, 0
04B0 72 15                      jc      short JUST_RETURN_5             ; Error waiting for REQ
04B2 24 0E                      and     al, 1110b
04B4 3C 0E                      cmp     al, 1110b                       ; BUSY | C/D | I/O
04B6 75 0E                      jnz     short NOT_GOOD                  ; Jump if Not Zero (ZF=0)
04B8 EC                         in      al, dx                          ; Input data
04B9 8A E0                      mov     ah, al

04BB 42                         inc     dx                              ; Port now 0x321
04BC                    WAIT_FOR_NOT_BUSY:
04BC EC                         in      al, dx                          ; Read Status
04BD 24 08                      and     al, 1000b                       ; BUSY bit
04BF 75 FB                      jnz     short WAIT_FOR_NOT_BUSY

04C1 86 C4                      xchg    al, ah
04C3 A8 02                      test    al, 10b                         ; ERROR bit
04C5 C3                         ret
04C6                    NOT_GOOD:
04C6 F9                         stc
04C7                    JUST_RETURN_5:
04C7 C3                         ret
04C8                    CHECK_STATUS    ENDP

                        ;--------------------------------------------------------
                        ; Wait for the controller to be ready for data          :
                        ; transmission in either way.                           :
                        ;                                                       :
                        ; Return:                                               :
                        ;   DX = 320H                                           :
                        ;   AL = Status (from port 321H)                       :
                        ;   Error 80H (timeout) on failure                      :
                        ;--------------------------------------------------------

04C8                    WAIT_FOR_REQ    PROC    NEAR
04C8 E8 0500 R                  call    PORT_321_TO_DX
04CB                    WAIT_FOR_REQ_WHILE_BUSY:
04CB EC                         in      al, dx                          ; Read Status
04CC A8 01                      test    al, 1                           ; REQ (Data available for reading)
04CE 75 07                      jnz     short GOT_REQ
04D0 A8 08                      test    al, 1000b                       ; BUSY
04D2 75 F7                      jnz     short WAIT_FOR_REQ_WHILE_BUSY
04D4 F9                         stc                                     ; Error: No longer BUSY,
                                                                        ;   but we have not seen REQ
04D5 B4 80                      mov     ah, INT_13H_ERR_DRIVE_TIMED_OUT

04D7                    GOT_REQ:
04D7 4A                         dec     dx                              ; Port is now 0x320 DATA
04D8 C3                         ret                                     ; Return Near from Procedure
04D9                    WAIT_FOR_REQ    ENDP
```

```
                        PAGE
                        ;--------------------------------------------------------
                        ; This picks the drive parameter table depending on     :
                        ; the drive selected and the jumper settings.          :
                        ;                                                      :
                        ; Returns                                              :
                        ;   ES = 00H                                           :
                        ;   BX = Drive parameters table (INT 41H)             :
                        ;--------------------------------------------------------

04D9                    GET_DRIVE_PARMS PROC    NEAR
04D9 33 C0                      xor     ax, ax
04DB 8E C0                      mov     es, ax
                                assume  es:ZERO
04DD 26: C4 1E 0104 R           les     bx, dword ptr es:INT_41H_OFF

04E2 E8 04FB R                  call    PORT_322_TO_DX
04E5 EC                         in      al, dx                          ; Read Jumpers

04E6 F6 06 0043 R 20            test    SCRATCH+1, 20h                  ; Is second drive selected?
04EB 75 04                      jnz     short PICKED_A_DRIVE            ; Jump if we're using the second drive
04ED D0 E8                      shr     al, 1                           ; Discard two least significant bits:
04EF D0 E8                      shr     al, 1                           ;   that would be the second drive.
04F1                    PICKED_A_DRIVE:

04F1 25 0003                    and     ax, 3
04F4 B1 04                      mov     cl, 4
04F6 D3 E0                      shl     ax, cl
04F8 03 D8                      add     bx, ax
04FA C3                         ret
04FB                    GET_DRIVE_PARMS ENDP

                        ;--------------------------------------------------------
                        ; Write: Select BUSY                                    :
                        ; Read: Jumpers                                        :
                        ;--------------------------------------------------------

04FB                    PORT_322_TO_DX  PROC    NEAR
04FB 8D 94 0322                 lea     dx, [si+322h]
04FF C3                         ret
0500                    PORT_322_TO_DX  ENDP

                        ;--------------------------------------------------------
                        ; Write: Reset                                          :
                        ; Read: Status                                         :
                        ;--------------------------------------------------------

0500                    PORT_321_TO_DX  PROC    NEAR
0500 8D 94 0321                 lea     dx, [si+321h]
0504 C3                         ret
0505                    PORT_321_TO_DX  ENDP

                        ;--------------------------------------------------------
                        ; Write: Set DMA/IRQ mask                               :
                        ;--------------------------------------------------------

0505                    PORT_323_TO_DX  PROC    NEAR
0505 8D 94 0323                 lea     dx, [si+323h]
0509 C3                         ret
050A                    PORT_323_TO_DX  ENDP

                        ;--------------------------------------------------------
                        ;       Sense byte to INT 13H error mapping (sparse)    :
                        ;--------------------------------------------------------

050A 09                 SENSE_TO_ERR    db SENSE_1X-SENSE_0X
050B 08                                 db SENSE_0X-SENSE_TO_ERR
050C 0A                                 db SENSE_2X-SENSE_1X
050D 11                                 db SENSE_1X-SENSE_TO_ERR
050E 02                                 db SENSE_3X-SENSE_2X
050F 1B                                 db SENSE_2X-SENSE_TO_ERR
0510 03                                 db SENSE_END-SENSE_3X
0511 1D                                 db SENSE_3X-SENSE_TO_ERR
0512 00                 SENSE_0X        db 0                                    ; 00H No Error Detected
0513 20                                 db INT_13H_ERR_CONTROLLER_FAILURE       ; 01H
0514 40                                 db INT_13H_ERR_SEEK_FAILURE             ; 02H No SC Signal
0515 20                                 db INT_13H_ERR_CONTROLLER_FAILURE       ; 03H Write Fault Signal
0516 80                                 db INT_13H_ERR_DRIVE_TIMED_OUT          ; 04H Drive Not Ready
0517 00                                 db 0                                    ; 05H
0518 20                                 db INT_13H_ERR_CONTROLLER_FAILURE       ; 06H Track 0 Not Found
0519 00                                 db 0                                    ; 07H
051A 40                                 db INT_13H_ERR_SEEK_FAILURE             ; 08H Drive Still Seeking
051B 10                 SENSE_1X        db INT_13H_ERR_CRC_ECC_DATA             ; 10H
051C 10                                 db INT_13H_ERR_CRC_ECC_DATA             ; 11H Uncorrectable Data Error
051D 02                                 db INT_13H_ERR_CANNOT_FIND_ADDR_MARK    ; 12H No Data Address Mark
051E 00                                 db 0                                    ; 13H
051F 04                                 db INT_13H_ERR_SECTOR_NOT_FOUND         ; 14H
0520 40                                 db INT_13H_ERR_SEEK_FAILURE             ; 15H Seek Error
0521 00                                 db 0                                    ; 16H
0522 00                                 db 0                                    ; 17H
0523 11                                 db INT_13H_ERR_ECC_CORRECTED_DATA       ; 18H Correctable Data error
0524 0B                                 db INT_13H_ERR_BAD_CYLINDER_OR_TRACK    ; 19H Track Is Flagged Bad
0525 01                 SENSE_2X        db INT_13H_ERR_INVALID_COMMAND          ; 20H Invalid Command
0526 02                                 db INT_13H_ERR_CANNOT_FIND_ADDR_MARK    ; 21H Illegal Sector Address
0527 20                 SENSE_3X        db INT_13H_ERR_CONTROLLER_FAILURE       ; 30H Sector Buffer Error
0528 20                                 db INT_13H_ERR_CONTROLLER_FAILURE       ; 31H BIOS ROM checksum error
0529 10                                 db INT_13H_ERR_CRC_ECC_DATA             ; 32H ECC Polynomial Error
052A                    SENSE_END       LABEL   NEAR

```

```
                        PAGE
                        ; ALIGN 16
052A    06 [            PAD             DB      6 DUP(0)
               00
                   ]


                        ;--------------------------------------------------
                        ;       Formatter routine                          :
                        ;                                                  :
                        ; Intended to be run from PC DOS debug with: -G=C800:5  :
                        ;--------------------------------------------------

0530                    DO_FORMAT       PROC    NEAR
0530 50                         push    ax
0531 8C C8                      mov     ax, cs
0533 8E D8                      mov     ds, ax
                                assume  ds:ROM

0535 BA 05D2 R                  mov     dx, offset STR_FORMAT_PROMPT
0538 B4 09                      mov     ah, 09H                         ; Print $-terminated string
053A CD 21                      int     21h

053C 58                         pop     ax
053D 0A C0                      or      al, al
053F 75 02                      jnz     short SECOND_DRIVE
0541 B0 03                      mov     al, 3

0543                    SECOND_DRIVE:
0543 80 E4 07                   and     ah, 7
0546 50                         push    ax
0547 80 C4 43                   add     ah, 'C'
054A 8A D4                      mov     dl, ah
054C B4 02                      mov     ah, 02H                         ; Print a character
054E CD 21                      int     21h

0550 BA 068F R                  mov     dx, offset STR_WITH_INTERLEAVE
0553 B4 09                      mov     ah, 09H                         ; Print $-terminated string
0555 CD 21                      int     21h

0557 58                         pop     ax
0558 50                         push    ax
0559 E8 05B7 R                  call    PRINT_NUMBER
055C B4 01                      mov     ah, 01H                         ; Get character
055E CD 21                      int     21h

0560 3C 79                      cmp     al, 'y'
0562 74 0C                      je      short FORMAT_YES_ANSWERED
0564 3C 59                      cmp     al, 'Y'
0566 74 08                      je      short FORMAT_YES_ANSWERED

0568 58                         pop     ax
0569 BA 06D0 R                  mov     dx, offset STR_NOTHING_DONE
056C 2A FF                      sub     bh, bh
056E EB 3D                      jmp     short PRINT_MESSAGE_AND_EXIT

0570                    FORMAT_YES_ANSWERED:
0570 58                         pop     ax
0571 8A D4                      mov     dl, ah
0573 80 CA 80                   or      dl, 80h
0576 8A E2                      mov     ah, dl
0578 50                         push    ax

0579 B4 12                      mov     ah, 12h                         ; Sector buffer diagnostics
057B CD 13                      int     INT_13H
057D 8A FC                      mov     bh, ah
057F 72 1A                      jc      short PRINT_ERROR_CODE_AND_EXIT

0581 58                         pop     ax
0582 50                         push    ax
0583 2A F6                      sub     dh, dh
0585 B9 0001                    mov     cx, 1
0588 B4 07                      mov     ah, 07H                         ; Format drive from track 1
058A CD 13                      int     INT_13H
058C 8A FC                      mov     bh, ah
058E 72 0B                      jc      short PRINT_ERROR_CODE_AND_EXIT

0590 B9 0001                    mov     cx, 1                           ; Recalibrate
0593 B4 11                      mov     ah, 11h
0595 CD 13                      int     INT_13H
0597 8A FC                      mov     bh, ah
0599 73 0F                      jnc     short PRINT_SUCCESS_MESSAGE_AND_EXIT

059B                    PRINT_ERROR_CODE_AND_EXIT:
059B 58                         pop     ax
059C BA 06B5 R                  mov     dx, offset STR_ERROR_COMPLETION
059F B4 09                      mov     ah, 09H                         ; Print $-terminated string
05A1 CD 21                      int     21h
05A3 8A C7                      mov     al, bh
05A5 E8 05B7 R                  call    PRINT_NUMBER
05A8 EB 07                      jmp     short DO_EXIT

05AA                    PRINT_SUCCESS_MESSAGE_AND_EXIT:
05AA BA 06A1 R                  mov     dx, offset STR_FORMAT_SUCCESS

05AD                    PRINT_MESSAGE_AND_EXIT:
05AD B4 09                      mov     ah, 09H                         ; Print $-terminated string
05AF CD 21                      int     21h

05B1                    DO_EXIT:
05B1 8A C7                      mov     al, bh                          ; Return value
05B3 B4 4C                      mov     ah, 4Ch                         ; Terminate
05B5 CD 21                      int     21h
05B7                    DO_FORMAT       ENDP

```

```
                        PAGE
05B7                    PRINT_NUMBER    PROC    NEAR

05B7 50                         push    ax
05B8 B1 04                      mov     cl, 4
05BA D2 E8                      shr     al, cl
05BC E8 05C3 R                  call    PRINT_DIGIT
05BF 58                         pop     ax
05C0 EB 01                      jmp     short PRINT_DIGIT
05C2 90                         nop
05C3                    PRINT_NUMBER    ENDP

05C3                    PRINT_DIGIT     PROC    NEAR

05C3 24 0F                      and     al, 0Fh
05C5 04 90                      add     al, 90h
05C7 27                         daa
05C8 14 40                      adc     al, 40h
05CA 27                         daa
05CB 8A D0                      mov     dl, al
05CD B4 02                      mov     ah, 02H                         ; Print a character
05CF CD 21                      int     21h
05D1 C3                         ret
05D2                    PRINT_DIGIT     ENDP

05D2                    STR_FORMAT_PROMPT:
05D2 57 58 32 20 46 6F          db      "WX2 Format Revision 6.0 "
     72 6D 61 74 20 52
     65 76 69 73 69 6F
     6E 20 36 2E 30 20
05EA 28 43 29 20 43 6F          db      "(C) Copyright Western Digital Corp. 1985",0Dh,0Ah
     70 79 72 69 67 68
     74 20 57 65 73 74
     65 72 6E 20 44 69
     67 69 74 61 6C 20
     43 6F 72 70 2E 20
     31 39 38 35 0D 0A
0614 20 20 20 28 41 48          db      "   (AH) = Relative drive number (0 - 7)",0Dh,0Ah
     29 20 3D 20 52 65
     6C 61 74 69 76 65
     20 64 72 69 76 65
     20 6E 75 6D 62 65
     72 20 28 30 20 2D
     20 37 29 0D 0A
063D 20 20 20 28 41 4C          db      "   (AL) = Interleave factor (3 is standard)",0Dh,0Ah
     29 20 3D 20 49 6E
     74 65 72 6C 65 61
     76 65 20 66 61 63
     74 6F 72 20 28 33
     20 69 73 20 73 74
     61 6E 64 61 72 64
     29 0D 0A
066A 50 72 65 73 73 20          db      'Press "y" to begin formatting drive $'
     22 79 22 20 74 6F
     20 62 65 67 69 6E
     20 66 6F 72 6D 61
     74 74 69 6E 67 20
     64 72 69 76 65 20
     24

068F                    STR_WITH_INTERLEAVE:
068F 20 77 69 74 68 20          db      " with interleave $"
     69 6E 74 65 72 6C
     65 61 76 65 20 24

06A1                    STR_FORMAT_SUCCESS:
06A1 0D 0A 46 6F 72 6D          db      0Dh,0Ah,"Format Successful$"
     61 74 20 53 75 63
     63 65 73 73 66 75
     6C 24

06B5                    STR_ERROR_COMPLETION:
06B5 0D 0A 45 72 72 6F          db      0Dh,0Ah,"Error---completion code $"
     72 2D 2D 2D 63 6F
     6D 70 6C 65 74 69
     6F 6E 20 63 6F 64
     65 20 24

06D0                    STR_NOTHING_DONE:
06D0 0D 0A 4E 6F 74 68          db      0Dh,0Ah,"Nothing Done Exit$"
     69 6E 67 20 44 6F
     6E 65 20 45 78 69
     74 24

06E4 38 35 2F 30 31 2F          db      "85/01/29"                      ; Date code
     32 39

06EC                    ROM     ends
                        END
```

```
;-------------------------------------------------------
;                                                      :
; This has been disassembled and commented manually on  :
; Sun 06 Jun 2021 in hope it will be useful to          :
; somebody. When assembled, it matches the              :
; 62-000042-012 ROM byte-for-byte.                      :
;                                                      :
; Please note that it is not free software and is       :
; subject to copyright.                                 :
;                                                      :
;---------------------------------------- SMOKE WEED --
```